

## Description

# [INDICATING NETWORK RESOURCE AVAILABILITY METHODS, SYSTEM AND PROGRAM PRODUCT]

### BACKGROUND OF INVENTION

[0001] Technical Field

[0002] The present invention relates generally to client-server environments, and more particularly, to methods, system and program product for indicating the availability of a network resource.

[0003] Related Art

[0004] Referring to FIG. 1, in a networked computer system 2 where individual client systems 4 must access a network resource 6, problems are encountered when a client system 4 attempts to access network resource 6 when it is unavailable. Unavailability can be caused by a myriad of situations such as the network resource, or a part thereof, being shut down or locking. Unavailability results in long

timeout intervals in which access to network resource 6 is blocked. For example, where network resource 6 is a server, when the server's file directory is unavailable, the server is generally unavailable. In these situations, a client system 4 application may 'hang up' as it tries to interface with the network resource. This 'hang up' creates unexpected delays that can result in problems such as a user assuming client system 4 has crashed or actual application lockage. In one example, delays of several minutes are incurred when a client system application is seeking to interface with a file server with an unavailable file directory.

[0005] Known solutions to this problem include tracking system management events through a system management interrupt (SMI) handler. SMI handlers operate within the network resource to react to problems, and may log events in memory or a system register. In another approach, an SMI handler may notify applications of a change in network resource status. However, this approach is generally limited to system recognized "system management" issues, such as the opening of a panel or a temperature excursion. Problems related to the availability of network-accessible file systems not related to lower level hardware

issues are likely not to be detected in real-time by system management monitoring. In another approach, various applications are offered that can poll the network resource register or file systems to determine if there has been a change in system status. These solutions, however, are limited as polling from a multitude of client systems in a networked environment expends resources necessary for other functions.

[0006] In view of the foregoing, there is a need in the art for a way to determine network resource availability in order to avoid time delays caused by querying unavailable resources.

#### **SUMMARY OF INVENTION**

[0007] The invention includes methods, system and program product for determining network resource availability in order to avoid time delays caused by querying unavailable resources. In one embodiment, the invention queries the network resource to determine network resource availability and response time, and indicates the network resource is available only if the network resource is determined to be available within a specified response time.

[0008] A first aspect of the invention is directed to a method for indicating availability of a network resource in a client-

server environment, the method comprising the steps of: querying the network resource to determine network resource availability and response time, and setting a status indicator with the network resource availability; in the case that the status indicator indicates that the network resource is available, determining whether a query of the network resource is occurring; and in the case that a query is occurring, indicating the network resource is available only if an expanded time of the query is less than a specified response time.

[0009] A second aspect of the invention is directed to a system for indicating availability of a network resource in a client-server environment, the system comprising: a query component that determines network resource availability and response time; and a status component that communicates the network resource availability as available only if the network resource is available within a specified response time.

[0010] A third aspect of the invention is directed to a computer program product comprising a computer useable medium having computer readable program code embodied therein for indicating availability of a network resource in a client-server environment, the program product com-

prising: program code configured to query the network resource to determine network resource availability and response time, and set a status indicator with the network resource availability; and program code configured to indicate a status of the network resource by: determining, in the case that the status indicator indicates that the network resource is available, whether a query of the network resource is occurring; and indicating, in the case that a query is occurring, the network resource is available only if an expanded time of the query is less than a specified response time.

[0011] A fourth aspect of the invention is directed to a method for indicating availability of a network resource in a client-server environment, the method comprising the steps of: querying the network resource to determine network resource availability and response time; and indicating the network resource is available only if the network resource is determined to be available within a specified response time.

[0012] A fifth aspect of the invention is directed to a system for deploying a network resource availability indication application, comprising: a computer infrastructure being operable to: receive a query from a client regarding the avail-

ability of the network resource; query the network resource to determine network resource availability and response time; set a status indicator with the network resource availability; determining whether a query of the network resource is occurring in the case that the status indicator indicates that the network resource is available; and provide an indication to the client that the network resource is available only if an expanded time of the query is less than a specified response time in the case that a query is occurring.

[0013] The foregoing and other features of the invention will be apparent from the following more particular description of embodiments of the invention.

#### **BRIEF DESCRIPTION OF DRAWINGS**

[0014] The embodiments of this invention will be described in detail, with reference to the following figures, wherein like designations denote like elements, and wherein:

[0015] FIG. 1 shows a block diagram of a prior art client-network resource environment.

[0016] FIG. 2 shows a block diagram of a client-network resource environment according to the invention including a network resource availability indicating system.

[0017] FIGS. 3A-3C show flow diagrams illustrating a first em-

bodiment of operational methodology of the system of FIG. 2.

#### **DETAILED DESCRIPTION**

[0018] The invention includes methods, system and program product that allow client system applications in a networked system to determine the availability of a network resource such as a server's file system, in order to avoid time delays caused by querying unavailable resources. In one embodiment, the invention implements a single system on the network to query a network resource in the form of a server at regular intervals to determine whether the server is available. However, the invention is applicable relative to a variety of other network resources.

[0019] The results of the querying are stored in memory on this system and made available, e.g., via TCP/IP, to client systems. All other systems on the network may query this system to quickly determine the state of the servers before attempting to negotiate a connection and transfer files from or to a server. By storing the status of network resources on a single system on the network and dedicating TCP/IP ports to providing the status of the server, networked systems do not need to individually query the network resource to determine availability. Furthermore, by

providing the information about the availability of the file systems on the server at the TCP/IP level of the polling system, the status of the server can be communicated to the networked systems on request more quickly than checking file systems.

[0020] With reference to the accompanying drawings, FIG. 2 is a block diagram of a network resource availability indicating system 10 in accordance with the invention. Client systems 104 query system 10 to determine the availability of a network resource 106, as will be described below. System 10 includes a memory 12, a processing unit (PU) 14, input/output devices (I/O) 16 and a bus 18. A database 20 may also be provided for storage of data relative to processing tasks. Memory 12 includes a program product 22 that, when executed by PU 14, comprises various functional capabilities described in further detail below. Memory 12 (and database 20) may comprise any known type of data storage system and/or transmission media, including magnetic media, optical media, random access memory (RAM), read only memory (ROM), a data object, etc. Moreover, memory 12 (and database 20) may reside at a single physical location comprising one or more types of data storage, or be distributed across a plurality of physical

systems. PU 14 may likewise comprise a single processing unit, or a plurality of processing units distributed across one or more locations. A server computer typically comprises an advanced mid-range multiprocessor-based server, such as the RS6000 from IBM, utilizing standard operating system software, which is designed to drive the operation of the particular hardware and which is compatible with other system components and I/O controllers. I/O 16 may comprise any known type of input/output device including a network system, modem, keyboard, mouse, scanner, voice recognition system, CRT, printer, disc drives, etc. Additional components, such as cache memory, communication systems, system software, etc., may also be incorporated into system 10.

[0021] As shown in FIG. 2, program product 22 may include a query component 24, a status component 26 and other system components 28. Other system components 28 may include any other functions required for operation of system 10 not expressly described herein. Query component 24 may include a query indicator 30 and an average query completion determinator 34. Status component 26 may include a status determinator 40 and a status indicator 42. The functions of the above components will be de-

scribed below.

[0022] Turning to FIGS. 3A–3B in conjunction with FIG. 2, a flow diagram illustrating a first embodiment of a method for indicating availability of a network resource in a client–server environment of system 10 is shown. Steps S1–S8 represent a querying loop that queries the availability of network resource 106 to determine network resource availability and response time, and steps S9–S16 represent a status loop that determines availability and indicates an availability status to a number of client systems 104. Each loop may function simultaneously.

[0023] Referring to FIG. 3A, in a first step S1, query component 24 begins a query of network resource 106 by setting a query indicator 30 to YES, which indicates that a query is occurring. At step S2, query component 24 initiates a read of network resource 106, e.g., a server, at a time 0. The term "read" is used to indicate the function relative to a network resource in the form of a server. It should be recognized that the actual query function, e.g., read, write, poll, etc., may vary depending on the type of network resource in question. At step S3, the query check completes at a time  $0 + n$ , where  $n$  is a time duration. At step S4, query component 24 sets query indicator to NO, which in–

dicates that a query is not presently occurring.

[0024] In step S5, query component 24 determines whether the function, e.g., read, was successful. At steps S6–S7, query component 24 sets a status indicator 30 with the network resource availability. If the function was unsuccessful, i.e., NO at step S5, then at step S6, query component 24 sets status indicator 42 to DOWN, which indicates network resource 106 is unavailable. Subsequently, processing returns to step S1 to monitor when network resource 106 becomes available again. That is, the querying steps repeat querying the network resource in the case that the network resource is unavailable. If the function was successful, i.e., YES at step S5, then at step S7, query component 24 sets status indicator 42 to available or UP, which indicates network resource 106 is available.

[0025] Next, at step S8, average query completion time determinator 34 updates an average query completion time using the query duration time  $n$ . The average query completion time is generated over a number of queries. A sleep period may follow step S8 prior to a new query beginning.

[0026] Turning to FIG. 3B in conjunction with FIG. 2, the status loop starts at step S9 with a client system 104 connecting to system 10 and requesting a status of network resource

106. In one embodiment, a client system 104 seeking information on network resource 106 (e.g., server "X") will contact a TCP/IP port for that network resource (e.g., "Xport") on system 10, rather than negotiating a contact and accessing network resource 106, e.g., file directories of a server. At step S10, status component 26 checks the availability of network resource 106 by checking status indicator 42. If the network resource is unavailable, i.e., NO at step S11, then processing proceeds to indicate the network resource is unavailable, i.e., DOWN, at step S16. In one embodiment, status component 26 may make this indication via a TCP/IP port ("Xport") by responding with either a "1" or a "0" indicating whether network resource 106 (e.g., server "X") is available/up or unavailable/down. In any event, system 10 increases the speed at which network resource status can be determined by an application.

[0027] In the case that status indicator 42 indicates that network resource 106 is available, i.e., YES at step S11, then at step S12, status component 26 determines whether a query of network resource is occurring by checking query indicator 30. That is, whether steps S1–S8 (FIG. 3A) are currently querying network resource 106 and query indicator 30 accordingly indicates that with 'YES.' In the case

that a query is not occurring, i.e., query indicator 30 is set to NO, then status component indicates that network resource is available, i.e., UP, at step S14, to a client.

[0028] In the case that a query is occurring, i.e., YES at step S13, status component 26 indicates network resource 106 is available to a client 104 only if an expanded time of the query is less than a specified response time, as determined at step S15. In particular, status determinator 40 (FIG. 2) determines whether an expanded time of the currently occurring query is longer than a specified response time. In this fashion, if a query is taking an exceptionally long period to complete in the querying loop S1–S8, status component 26 can indicate the network resource is unavailable even if the querying loop has not reached that conclusion. In one embodiment, the specified response time is defined as a value within a threshold of the average query completion time, e.g., the average successful completion time plus some multiple of the standard deviation of successful completion times. That is, status component 26 indicates the network resource 106 is unavailable if a currently occurring query is taking an unacceptably long time, when compared with the average query completion time.

[0029] In the case that the expanded time is less than the specified response time, i.e., YES at step S15, then processing proceeds to step S14 at which status component 26 indicates that network resource 106 is available, i.e., UP, to a client. Alternatively, if the expanded time frame is not less than the specified response time, i.e., NO at step S15, then status component 26 indicates the network resource 106 is unavailable, i.e., DOWN, at step S16.

[0030] Turning to FIG. 3C, as an alternative embodiment, if a client accesses the network resource 106 and determines that network resource 106 is unavailable, the client 104 may set status indicator 42 to unavailable, i.e., DOWN. In this fashion, the most up-to-date information regarding network resource 106 availability can be maintained, even in the event of system 10 falsely reporting network resource 106 as available. Processing then proceeds as indicated above.

[0031] The above-described processing can occur at the initiation of each transaction requested by a client. However, in a preferred embodiment, the process can occur at the beginning of each step of a transaction to prevent a transaction from starting, and then being delayed if the network resource becomes unavailable.

[0032] While the above-described system 10 has been illustrated as a separate system, it should be recognized that system 10 may be implemented as part of a client system 104. However, this implementation has disadvantages relative to resource usage on client system 104.

[0033] Once the availability of network resource 106 has been ascertained, client system 104 can proceed to take appropriate action. That is, if network resource 106 is available, client system 104 can access the network resource. If network resource 106 is unavailable, client system 104 can avoid an indefinite delay by not accessing network resource 106 and is able to take immediate action, for instance, return a fail-code to the application, or a message to the user.

[0034] In the previous discussion, it will be understood that the method steps discussed are performed by a processor, such as PU 14 of system 10, executing instructions of program product 22 stored in memory. It is understood that the various devices, modules, mechanisms and systems described herein may be realized in hardware, software, or a combination of hardware and software, and may be compartmentalized other than as shown. They may be implemented by any type of computer system or

other apparatus adapted for carrying out the methods described herein. A typical combination of hardware and software could be a general-purpose computer system with a computer program that, when loaded and executed, controls the computer system such that it carries out the methods described herein. Alternatively, a specific use computer, containing specialized hardware for carrying out one or more of the functional tasks of the invention could be utilized. The present invention can also be embedded in a computer program product, which comprises all the features enabling the implementation of the methods and functions described herein, and which – when loaded in a computer system – is able to carry out these methods and functions. Computer program, software program, program, program product, or software, in the present context mean any expression, in any language, code or notation, of a set of instructions intended to cause a system having an information processing capability to perform a particular function either directly or after the following: (a) conversion to another language, code or notation; and/or (b) reproduction in a different material form. It should also further be understood that the present invention could be offered as a business method

whereby users are charged on a fee or subscription basis.

[0035] While this invention has been described in conjunction with the specific embodiments outlined above, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, the embodiments of the invention as set forth above are intended to be illustrative, not limiting. Various changes may be made without departing from the spirit and scope of the invention as defined in the following claims.